MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2017/2018

ECP4256 –PRACTICAL FPGA DESIGN AND INTERFACING

(TE / RE)

6 MARCH 2018
9.00 a.m. - 11.00 a.m.
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This Question paper consists of 9 pages with 5 Questions only.

2. Attempt **ALL** the questions. All questions carry equal marks and the distribution of the marks for each question is given.

3. Please write all your answers in the Answer Booklet provided.

Question 1

a)

(i) Complete the truth table of the full adder as shown in Figure-1(a). [4 marks]

(ii) Obtain the simplified expression for Sum (S) and Carryout (Cout). [2 marks]

(iii) Implement the full adder using Programmable Read only Memory (PROM). [4 marks]

Inputs			Output	
A	B	Cin	S	Cout
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Figure-1(a) Full adder truth table

b) The Altera Cyclone II FPGA Logic Array Block (LAB) architecture is shown in Figure-1(b). Describe any **FIVE(5)** features of LAB. [10 marks]

Continued ...

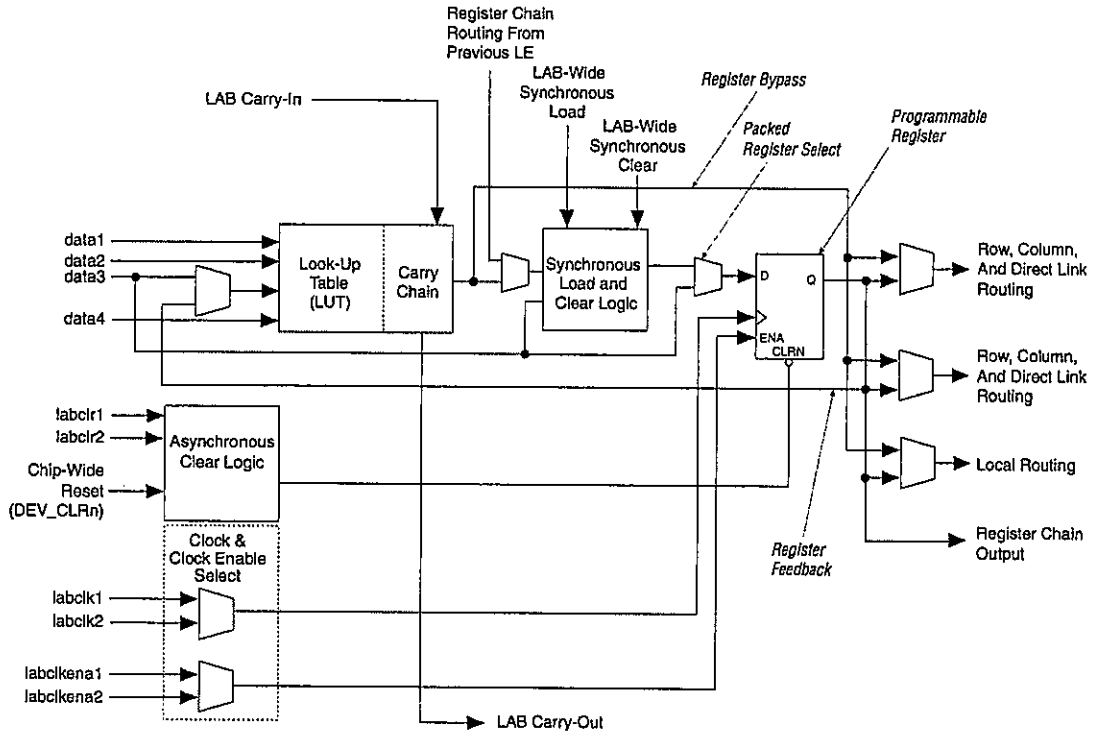

Figure-1(b) Altera Cyclone II FPGA Logic Array Block Architecture

Continued ...

Question 2

a) An 8-function Arithmetic and Logic Unit (ALU) is designed using CASE construct as shown in Figure-2. There are two 4-bit inputs: operand A [3:0] and B [3:0] and one 3-bit input opcode op [2:0]. There is one 8-bit output Result [7:0] which contains the result of the operations and is declared to accommodate the 2n-bit product. The operation codes are shown in Table-2. Write the Verilog description of the ALU. [12 marks]

b) Write the testbench to verify the functionality of the ALU. [8 marks]

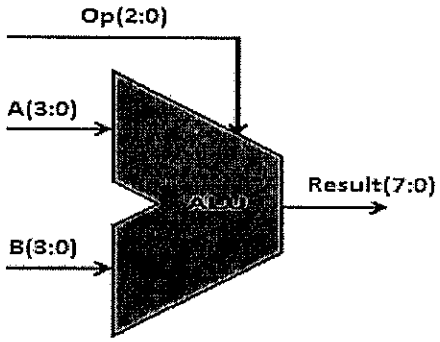

Figure-2 8-function Arithmetic and Logic Unit (ALU)

Table-2 Operation codes for 8-function Arithmetic and Logic Unit (ALU)

Opcode	Operation	Mnemonic
000	Add	add_op
001	Subtract	sub_op
010	Multiply	mul_op
011	Divide	div_op
100	AND	and_op
101	OR	or_op
110	Exclusive-OR	xor_op
111	Exclusive-NOR	xnor_op

Continued ...

Question 3

a) Explain the concepts with ASM chart.

(i) chaining [3 marks]

(ii) multi-cycling [3 marks]

(iii) The addition and multiplication operations are mapped on to an adder and multiplier with a 40ns and 80ns propagation delay, respectively. If the control step length is set to the 80ns (the longest of the two propagation delays), and each operation is scheduled into exactly one control step, the overall length of the schedule will be 160ns, and the multiplier will be utilized only half of the time as shown in Figure-3.

Explain how the multiplier utilization can be increased by using following optimization techniques with neat diagram.

1. Chaining
2. Multi-cycling [10 marks]

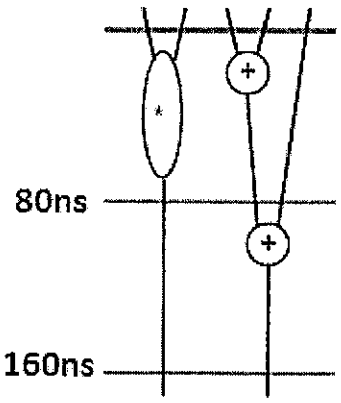

Figure-3 without chaining and multi-cycling

Continued ...

b) Discrete Cosine Transform (DCT) consists of two matrix multiplications which are computationally identical. The DCT code in C is shown in Table-3. Discuss the effects of process pipelining when it is applied to DCT. [4 marks]

Table-3 DCT code

```
void DCT (int InBlock[][8], int OutBlock[][8]) {
  int TempBlock[8][8], CosTrans[8][8];
  Transpose(CosBlock, CosTrans);

 /* TempBlock = InBlock * CosBlock^T */
MatrixMult(InBlock,CosTrans,TempBlock);

/* OutBlock = CosBlock * TempBlock */
MatrixMult(CosBlock,TempBlock,OutBlock);
}
```

Question 4

Design a state machine controller for the state diagram as shown in Figure-4(a). The controller will read a binary data from toggle switch and display it on a seven segment display.

The controller will read input **go** to decide to start/halt the operation. Output LED **on_led** indicates that the controller is reading/displaying data. Output LED **rst_led** indicates that the controller is in RESET state. The datapath function of the controller is given in Table-4(a). The kick-start code is given in Table-4(b). Complete the state machine controller design using the kick-start code. [20 marks]

Continued ...

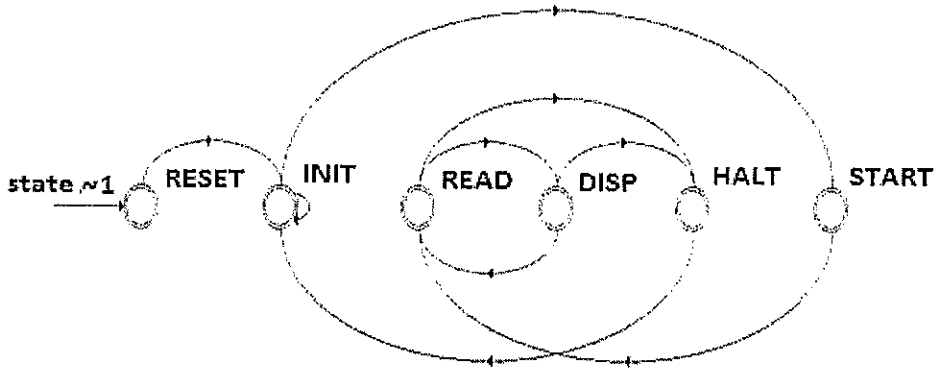

Figure-4(a) State machine Diagram

Table-4(a) Datapath function of the controller

State	Datapath
RESET	Reset all registers -Turn on **rst_led**
INIT	Initialize registers -Turn off **rst_led**
START	Read input go • **go = 0**, start the read/display operation, turn on **on_led** • **go = 1**, wait
READ	Read data
DISP	Write to seven segment display
HALT	Turn off **on_led**

Continued ...

Table-4(b) kick-start code

```
module Q4 (
input clk, rst, go,
input [3:0] data,
output on_led, rst_led,
output [6:0] HEX0,
output [7:0] state_disp );

parameter RESET = 8'b00000001;
parameter INIT  = 8'b00000010;
parameter START = 8'b00000100;
parameter READ  = 8'b00001000;
parameter DISP  = 8'b00010000;
parameter HALT  = 8'b00100000;

wire clk_1_hz;
reg [7:0] state=RESET, next_state;
reg [3:0] disp, BCD;
reg on_off, reset;

// Your state machine controller here

Clock_1_Hz clk1(
.clk_in(clk),
.clk_out(clk_1_hz));
BCD2SevenSegment dig0(
.BCD(BCD),
.HEX(HEX0));
assign state_disp = state;
assign on_led = on_off;
assign rst_led = reset;
endmodule
```

Continued ...

Question 5

a)

(i) What is System On a Programmable Chip (SOPC)? [2.5 marks]

(ii)Compare the SOPC Design versus Traditional Design(ASIC and Fixed Processor) Modalities in terms of

i. Hardware Flexibility
ii. Reconfigurability
iii. Production cost
iv. Power efficiency
v. Performance

[7.5 marks]

b) Design a SOPC system on Altera cyclone II FPGA chip for a production line controller with the following features.

i. Two interfaces to control conveyer belt motor driver and actuators driver
ii. To read the data from sensors
iii. An indicator (LED) to indicate the output
iv. Switches to read from the push buttons

Note: Use a block Diagram to illustrate your design and describe the functionality of each block. [10 marks]

End of Paper